

National University of Computer and Emerging Sciences



Lab Manual 10 Object Oriented Programming

Course Instructor	Ms. Syeda Tayyaba Bukhari
Lab Instructor (s)	Ms. Fariha Maqbool Mr. Muhammad Usama
Section	B
Semester	Spring 2022

Department of Computer Science
FAST-NU, Lahore, Pakistan

Objectives

After performing this lab, students shall be able to:

- ✓ Polymorphism
- ✓ Virtual functions (Function overriding)
- ✓ Abstract Base class(s) and Pure Virtual function(s)

TASK 1:

Design a Ship class that has the following members:

- A member variable for the name of the ship (a string)
- A member variable for the year that the ship was built (a string)
- A constructor and appropriate accessors and mutators
- A virtual print function that displays the ship's name and the year it was built.

Design a CruiseShip class that is derived from the Ship class. The CruiseShip class should have the following members:

- A member variable for the maximum number of passengers (an int)
- A constructor and appropriate accessors and mutators
- A print function that overrides the print function in the base class. The CruiseShip class's print function should display only the ship's name and the maximum number of passengers.

Design a CargoShip class that is derived from the Ship class. The CargoShip class should have the following members:

- A member variable for the cargo capacity in tonnage (an int).
- A constructor and appropriate accessors and mutators.
- A print function that overrides the print function in the base class. The CargoShip class's print function should display only the ship's name and the ship's cargo capacity.

Demonstrate the classes in a program that has an array of Ship pointers. The array elements should be initialized with the addresses of dynamically allocated Ship, CruiseShip, and CargoShip objects. The program should then step through the array, calling each object's print function.

TASK 2:

Define a pure abstract base class called BasicShape. The BasicShape class should have the following members:

Private Member Variable:

area, a double used to hold the shape's area.

Public Member Functions:

getArea. This function should return the value in the member variable area.

calcArea. This function should be a pure virtual function.

Next, define a class named Circle. It should be derived from the BasicShape class. It should have the following members:

Private Member Variables:

centerX, a long integer used to hold the x coordinate of the circle's center.

centerY, a long integer used to hold the y coordinate of the circle's center.

radius, a double used to hold the circle's radius.

Public Member Functions:

constructor—accepts values for centerX, centerY, and radius. Should call the overridden calcArea function described below.

getCenterX—returns the value in centerX.

getCenterY—returns the value in centerY.

calcArea—calculates the area of the circle ($\text{area} = 3.14159 * \text{radius} * \text{radius}$) and stores the result in the inherited member area.

Next, define a class named Rectangle. It should be derived from the BasicShape class. It should have the following members:

Private Member Variables:

width, a long integer used to hold the width of the rectangle.

length, a long integer used to hold the length of the rectangle.

Public Member Functions:

constructor—accepts values for width and length. Should call the overridden calcArea function described below.

getWidth—returns the value in width.

getLength—returns the value in length.

calcArea—calculates the area of the rectangle ($\text{area} = \text{length} * \text{width}$) and stores the result in the inherited member area.

After you have created these classes, create a driver program that defines a Circle object and a Rectangle object. Demonstrate that each object properly calculates and reports its area.